**Furniture Bazaar: Day 3 - API Integration and Data Migration**

# Project Overview

Project Name: Furniture Bazaar  
Goal: Create a marketplace for furniture with dynamic data display by integrating APIs and managing data migration.

# API Integration

## API Details

* API URL: https://template-0-beta.vercel.app/api/product
* Endpoint Used: /products

## Fetched Data Fields

1. name
2. description
3. price
4. discountPercentage
5. isFeaturedProduct
6. stockLevel
7. category
8. imageURL

### Security Measures

* **Sensitive Information:** API keys are securely stored using .env files to avoid exposure.

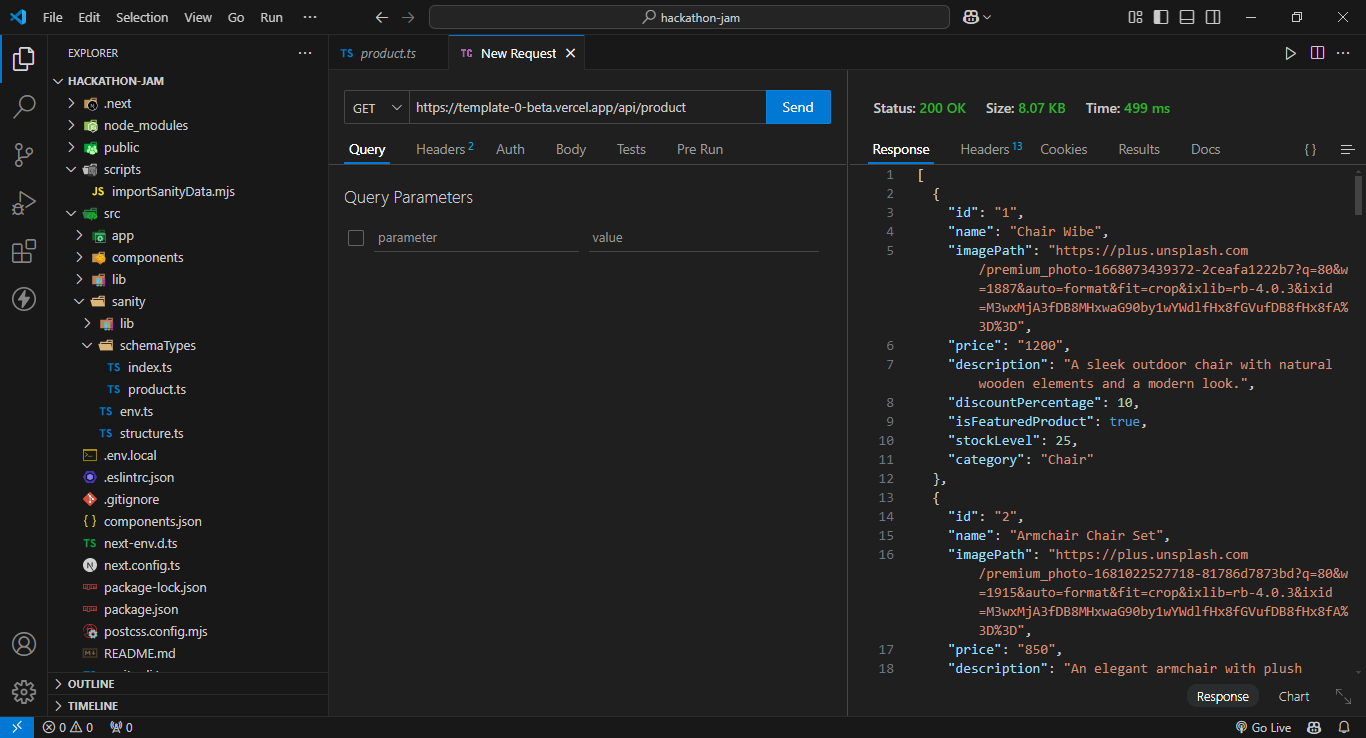
# Data Migration

## Existing Dataset

* Source: .JSON files.

### Validation

* **Method:** Schema validation ensures the integrity and correctness of data.



### Migration Method

* **Approach:** Data will be migrated using API fetching and scripting to automate the process and avoid manual errors while making importSanityData.mjs file in scripts folder and implementing code to fetch data to sanity.
* Install axios to fetch data **npm install @sanity/client axios**



* After writing the code we run the code to transfer data to sanity fields npm run importSanityData.mjs .

## Sanity Schema Adjustments

* Adjustments Needed: The schema will be modified to align with API data fields and maintain compatibility with frontend requirements.

### Critical Fields and Relationships:

* + Product fields such as name, price, and category will be mapped for proper display in the frontend.

# Frontend Integration

## Framework

* Technology Used: Next.js

## Display Strategy

* Implementation: The data will be dynamically displayed on the product listing page, fetched from Sanity CMS.

# Testing and Validation

### Tools

* Testing Platform: Thunder Client in VS Code is used to verify API endpoints and validate data migration scripts.

## Challenges and Edge Cases

* Handling cases where data fields like imageURL or category are missing.
* Ensuring smooth frontend updates when API data changes.

## Expected Deliverables

1. Successfully imported API data into Sanity CMS.
2. Dynamic data rendering in the frontend using Next.js and Sanity.

## Day 3 Checklist:

### Self Validation Checklist: Api Understanding: • ✔

**Schema Validation:**  
• ✔

**Data Migration:**

• ✔

Api Integration in Next.js:

• ✔

**Attachments**

**Sanity Schema Screenshot**: 

1. **Fronted Output**:  
   